Special Issue Article

JDMS

Journal of Defense Modeling and Simulation: Applications, Methodology, Technology
2017, Vol. 14(3) 201–216
© The Author(s) 2016
DOI: 10.1177/1548512916662924
journals.sagepub.com/home/dms
SAGE

Quantifying the mission impact of network-level cyber defensive mitigations

Neal Wagner, Cem Ş Şahin, Michael Winterrose, James Riordan, Diana Hanson, Jaime Peña and William W Streilein

Abstract

Modern missions of government and private organizations rely on computer networks to operate. As evidenced by several well-publicized cyber breaches, these missions are under attack. Several cyber defensive measures have been proposed to mitigate this threat, some are meant to protect individual hosts on the network, and others are designed to protect the network at large. From a qualitative perspective, these mitigations seem to improve security, but there is no quantitative assessment of their effectiveness with respect to a complete network system and a cyber-supported mission for which the network exists. The purpose of this paper is to examine network-level cyber defensive mitigations and quantify their impact on network security and mission performance. Testing such mitigations in an live network environment is generally not possible due to the expense, and thus a modeling and simulation approach is utilized. Our approach employs a modularized hierarchical simulation framework to model a complete cyber system and its relevant dynamics at multiple scales. We conduct experiments that test the effectiveness of network-level mitigations from the perspectives of security and mission performance. Additionally, we introduce a novel, unified metric for mitigation effectiveness that takes into account both of these perspectives and provides a single measurement that is convenient and easily accessible to security practitioners.

Keywords

Cyber defense, hierarchical modeling, mission impact, multi-scale systems, network security, defensive mitigation

1 Introduction

Cyber attacks are increasing at an alarming rate.[1,2] As exhibited by a number of high-profile cyber breaches,[3,4] the damage that these attacks can cause is substantial. To counter this threat, agencies such as the SANS Institute, Google, Microsoft, and the Information Assurance Directorate of the National Security Agency, among others, have proposed several cyber defensive mitigations.[5–8] Some of these mitigations are meant to protect at the host level via security controls deployed on individual network devices, while others are designed to protect at the network level via security controls deployed to the network at large. Given sufficient resources, network administrators and security practitioners could deploy all recommended mitigations to maximize a network's security posture. Unfortunately, the reality for most practitioners is one in which allotted security resources are limited, and thus they must choose mitigations that will provide the most security benefit for their network. Proposed mitigations, however, have not been quantitatively assessed for effectiveness and, consequently, are not ranked or prioritized in any way. This forces practitioners to rely on their own judgement to select appropriate mitigations.

Another salient point is that networks do not exist for their own sake, but rather exist to support an organizational mission. This means that practitioners must consider

Massachusetts Institute of Technology Lincoln Laboratory, USA

Corresponding author:
Neal Wagner, Massachusetts Institute of Technology Lincoln Laboratory, 244 Wood Street, Lexington, MA 02420, USA.
Email: neal.wagner@ll.mit.edu

tradeoffs between security and mission performance. Accordingly, mitigations should be examined in the context of a complete network system where both security and mission impact are taken into account.

It is important to note that cyber systems, like social systems, include human actors and are thus stochastic in nature. For this reason, a network system under a given set of conditions (e.g., users, attackers, and defensive mitigations) will not always lead to the same outcome. Rather, the pairing of a network system and a set of environmental conditions that affect it will generate a distribution of outcomes that represent the range of possible results. When evaluating a mitigation for a particular network environment, it is therefore necessary to execute numerous tests in order to determine which outcomes are probable and which are improbable. Conducting a large number of network-scale tests in a live environment requires significant resources and is generally infeasible. We, thus, focus on a modeling and simulation approach due to its relative low cost.

This paper examines two proposed defensive mitigations designed to protect the network at large from cyber attack: (i) Segregation of Networks and Functions (SNF); and (ii) Limiting Workstation-to-Workstation Communication (LWC). The purpose is to provide a quantitative assessment of these network-level mitigations in the context of a complete network system and to consider mitigation effectiveness with respect to two fundamental network concerns, security and mission impact. To this end, a modularized hierarchical simulation framework is utilized to capture and integrate relevant dynamics at the sub-net/enclave and full network scales. We also describe a novel metric that combines results for security and mission performance into a single unified measure of mitigation effectiveness that is convenient and easily accessible to security practitioners and network analysts.

The rest of this paper is organized as follows: Section 2 discusses the current state of the practice with respect to the use of modeling and simulation in the cyber security domain; Section 3 describes the network-level mitigations examined; Section 4 provides details of the multi-scale hierarchical simulation model, including component models capturing cyber threat, defense, and mission; Section 5 gives metrics quantifying security and mission impact and describes our unified measure for mitigation effectiveness; Section 6 discusses our simulation experiments; and Section 7 concludes.

2 Domain characterization

Cyber systems contain a mix of computerized processes, hardware entities, and human actors in an environment that is constantly shifting. These complexities make it difficult to predict the effects that policy changes will have on a network system and the mission it is intended to support. The cyber security community is charged with recommending defensive measures to improve network security and mitigate the threat of cyber attack. Currently, these recommendations are put forth as security-related best practices (e.g., see Microsoft's Enterpise Security Best Practices[7]). It is important to note that, generally, these recommendations are made via the judgement of subject matter experts and are not based on empirical analysis of actual network tests. As mentioned in Section 1, the reason for this is that executing security-related tests at the network scale is oftentimes prohibitively expensive.

In response to this situation, the modeling and simulation community has generated a body of work that is focused on capturing and analyzing network systems with the intent of improving their security. The following section summarizes the current state of this work and discusses the contributions of this paper and its place within this greater body of cyber modeling and simulation research.

2.1 State of the practice

A large number of studies have used modeling and simulation (mod/sim) as a tool to improve the detection of network intrusions.[9–11] These studies focus on network situational awareness and use mod/sim to execute initial tests of newly proposed intrusion detection techniques before moving these techniques to the prototyping stage.

Another set of studies focuses on utilizing mod/sim for the purpose of investigating network security in the context of specific threats and corresponding defenses. A study combining discrete event simulation with meta-heuristic optimization to simulate network attacks and optimize network defenses is provided in Kiesling et al.[12] An agent-based model investigating cooperative botnet attacks and corresponding defenses is presented in Kotenko et al.[13] A Markov model is used to simulate worm attacks with simulation splitting techniques for efficient simulation of rare catastrophic network states in Masi et al.[14] A model built using OMNeT + + to simulate distributed denial-of-service attacks on networks is presented in Mina et al.[15] In Priest et al.[16] an agent-based model is used to evaluate the performance of candidate security techniques that rely on a moving target strategy to defend against cyber attack. In Toutonji et al.[17] and Yu et al.[18] epidemiological models are employed to simulate malware propagation over networks. An agent-based simulation examines the effectiveness of security policies seeking to mitigate the threat posed by unauthorized hardware on a network in Wagner et al.[19]

Another vein of research applies game theory to model the interactions between attack and defense. Some recent examples include Clark et al.[20] and Pawlick et al.[21] In

Clark et al.[20] a game-theoretic approach is applied to evaluate network IP address randomization strategies for their ability to confuse attackers trying to locate network devices to attack. In Pawlick et al.[21] games are used to model interactions between user devices and cloud-based systems that are under attack and sometimes controlled by the attacker.

This paper utilizes a mod/sim approach to examine the effectiveness of two widely known network-level cyber defensive mitigations. Our main contribution is to provide a quantitative assessment of these mitigations' effectiveness at the network scale with respect to two fundamental network concerns: security and mission impact. We also use a novel metric to combine results of these concerns into a single unified measure that is easily accessible to security analysts and practitioners.

3 Network-level cyber defensive mitigations

Cyber attacks have caused significant damage to enterprise networks in recent years. Quantifying the performance of defensive mitigations helps network administrators make better decisions to improve the security posture of networks against attack. This paper examines two defensive mitigations that seek to provide security at the network level, SNF and LWC.[8] Both of these mitigations attempt to thwart an attacker's ability to move within a network after he/she has gained initial entry to the network.

3.1 SNF mitigation

The SNF mitigation is concerned with partitioning a network into sections or segments to protect sensitive or valuable resources. Different cyber assets (e.g., hosts, servers, sub-nets) are used for different organizational functions (e.g., public-facing web services, financial transactions, human resource management, etc.) having differing sensitivity levels and security requirements. The idea is to segregate these different groups of cyber assets based on their function and restrict communications between the segregated groups. This is thought to improve security by hampering the ability of an attacker, who has already gained a foothold on the network, to traverse the network, spread compromise, and acquire further access to sensitive resources. Segregation is typically implemented by firewalls, network egress and ingress filters, application-level filters, and/or physical (hardware) infrastructure.[22]

3.2 LWC mitigation

The LWC mitigation picks up where SNF leaves off. The idea is to regulate communications at a higher granularity. LWC controls communications to a greater extent than SNF, in which even devices within the same organizational function may have limited communications (or be prevented from communicating outright). Here, the goal is to enforce the *principle of least privilege* and to allow communication privileges only when necessary for task execution. LWC is implemented by setting device-level firewall rules (e.g., Windows Firewall rules), disabling remote logon access to devices, and using private virtual LANs.[23]

Both mitigations are about partitioning a network into segments and controlling communications between segments and between segments and the Internet. We refer to an individual segment of a partitioned network as an *enclave*, which is a group of network devices with homogeneous reachability.

4 Multi-scale hierarchical model

We wish to quantitatively assess the effectiveness of the SNF and LWC mitigations in the context of a complete network system. For this purpose a multi-scale model to characterize dynamics at the enclave and network scales is employed. The complete model is modularized via a hierarchical framework in which enclave-scale dynamics (i.e., dynamics internal to a single enclave) are captured separately in a single model, and simulation results from this model are then used to inform a network-scale model. The model is informed by a proprietary testbed environment in which a partitioned network is captured at a coarse-grained level of abstraction where only the vulnerability level of individual network enclaves is measured. A graphical overview of the full hierarchical model is given in Figure 1. From the figure, the enclave model is parameterized by outputs from testbed experiments (see Section 4.2). Simulation runs are executed on this enclave model, results are aggregated, and these results are used to parameterize the network model (right of the figure), which captures an abstracted full network system with attack/

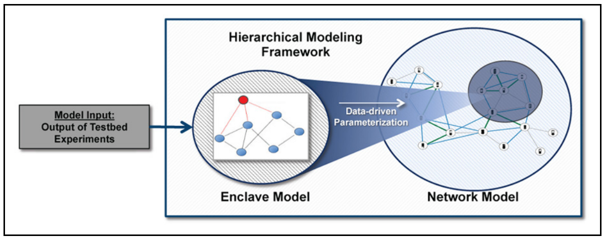

Hierarchical Modeling Framework
Model Input:
Output of Testbed Experiments
Data-driven Parameterization
Enclave Model
Network Model

Figure 1. Multi-scale, hierarchical model: Enclave model captures dynamics internal to a single enclave, network model captures network-scale dynamics of security and mission performance.

defense dynamics and mission users and their associated operations (details provided in Section 4.3).

With respect to this study, the hierarchical model structure is beneficial for the following reasons:

- it allows the model to incorporate data gleaned from testbed experiments;
- it provides for a reduction in model implementation effort due to the modularity gained by dividing the simulation model into multiple components; and
- it supports quicker simulation execution times due to reduced complexity at the larger scale network model.

With respect to future studies, this model structure provides a simulation framework that is flexible and re-usable: flexible because different versions of one component model can be substituted into the framework without having to change the underlying implementation of the other component model(s); re-usable because a component model may be used as part of multiple complete simulation models with potentially little or no modification. This study takes advantage of the framework's re-usability by re-tooling a network component model capturing mission users and operations from a previous study,[24] which itself was re-tooled from Priest et al.[16] Planned future work will take advantage of the framework's flexibility—it will focus on developing a simulation model to replace the testbed environment so that more partitioning scenarios can be easily examined, as the resource cost of executing scenarios on the testbed is relatively high. The following sections detail the components of the complete model and their integration.

4.1 Testbed environment

As discussed above, the testbed is a proprietary environment that supports coarse-grained tests of a partitioned network. A partitioning architecture that divides the network into enclaves and restricts communications between enclaves and between enclaves and the Internet can be instantiated. The environment modeled by the testbed is depicted in Figure 2. In this environment the attacker residing on the Internet is restricted by the partitioning architecture and can only communicate with enclaves as allowed by the architecture. For example, as shown in the figure, suppose a network is partitioned into three enclaves where Enclave 1 is allowed communication with the Internet and Enclaves 2 and 3 are not. Additionally, suppose communications between Enclave 1 and Enclave 3 are disallowed by the architecture. As displayed in the figure, the attacker can penetrate the network only through Enclave 1. If the attacker is successful at compromising Enclave 1 (indicated by the enclave's red color in the figure), then he/she can attempt to spread to Enclave 2 via the communication channel allowed by the architecture, but cannot spread directly to Enclave 3 because the architecture blocks communications between Enclaves 1 and 3. The testbed environment specifies communication channels by allowing or disallowing software services between enclaves. The environment also includes the notion of enclave cleansing by the defender (depicted in the upper right graphic of the figure): compromised enclaves are periodically cleansed and restored to an uncompromised state.

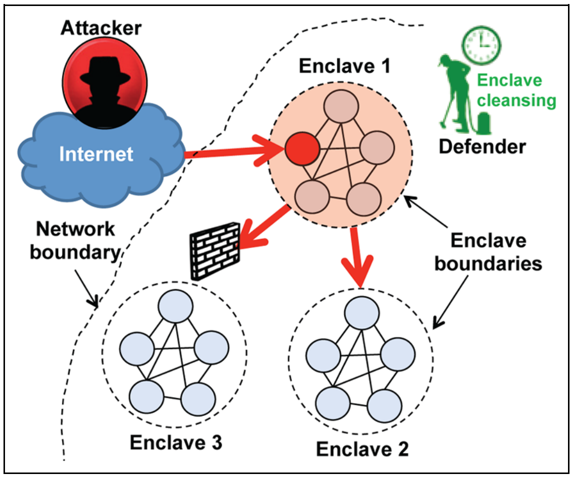


Figure 2. Testbed environment: Partitioning architecture divides network into enclaves and restricts communications between enclaves and between enclaves and the Internet. Attacker resides on the Internet and attempts to compromise enclaves via communication channels. Defender periodically cleanses compromised enclaves.

The testbed uses data from real software vulnerabilities and corresponding exploits to characterize the vulnerability level of individual enclaves with respect to a given network partitioning architecture and enclave-cleansing rate. The environment measures the probability that an enclave has been penetrated but does not capture instances of actual device compromise within an enclave. This measurement informs the enclave component model (depicted in Figure 1).

4.2 Enclave model

The enclave model seeks to characterize the dynamics of attack and defense, at the device level, within a single enclave. The threat model is that of an attacker who penetrates the enclave by compromising a single enclave device and attempts to spread to other enclave devices. This threat model is depicted in Figure 3.

An epidemic model is used to capture device-to-device infection spreading within an enclave. We utilize the

Algorithm 1: Enclave model

```
procedure Enclave($p_{vuln}$,$\beta$,$N$)   ▷ $p_{vuln}$: probability enclave is vulnerable, $\beta$: infection spread rate, $N$: no. of enclave devices
  repeat
    $t \leftarrow 0$
    $d_{comp} \leftarrow$ [empty set]   ▷ Set of compromised enclave devices
    $d_{uncomp} \leftarrow$ [all enclave devices]   ▷ Set of uncompromised enclave devices
    $r \leftarrow \mathbb{N}[0,1]$   ▷ $r$ is assigned a random value $\in [0,1]$
    $I(0) \leftarrow 1$
    while $r < p_{vuln}$ do   ▷ Enclave is vulnerable, infection spread can occur
      $I(t) \leftarrow f(I(0),\beta,N,t)$   ▷ Compute no. of infected devices using Equation (1)
      if $|d_{comp}| < I(t)$ then
        randomly remove $I(t) - |d_{comp}|$ devices from $d_{uncomp}$, add to $d_{comp}$
      $t \leftarrow t+1$
      $r \leftarrow \mathbb{N}[0,1]$
  until Total timesteps > Maximum timesteps
```

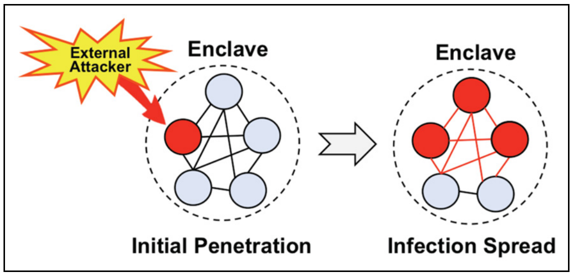


Figure 3. Enclave threat model: attacker compromises a single device and then spreads throughout enclave.

propagation model proposed by Yu et al.[18] and given by Equation (1):

$$I(t) = I(0) \times e^{\beta N t} \tag{1}$$

where t is time, $I(0)$ is the number of infected devices at $t = 0$, β is the infection propagation rate, N is the total number of devices in the enclave, and $I(t)$ computes the total number of infected devices at time t. Initial enclave penetration is modeled as compromise of a single device (i.e., $I(0) = 1$).

The defense model is an abstraction of the protection provided by the network partitioning architecture and enclave cleansing rate captured in the testbed environment but from the perspective of a single enclave. The model specifies a random variable to capture the probability that an enclave is in a vulnerable state (i.e., whether or not it has been penetrated by the attacker). When the enclave is vulnerable, infection can spread from device to device; when the enclave is not vulnerable (i.e., it has been cleansed by the defender), all enclave devices are uninfected. The full enclave model is given by Algorithm 1.

In Algorithm 1, the probability that an enclave is vulnerable, p_{vuln}, is specified by the output of testbed experiments for the enclave being modeled. The model generates three outputs that characterize the security of devices in the enclave: the expected number of devices that are compromised at any given moment, the mean duration time of compromise for devices when they are compromised, and the standard deviation of compromise duration times. As mentioned above, these outputs are used to inform a network-scale model (depicted in Figure 1 and detailed below).

4.3 Network model

The network model characterizes the dynamics of attack, defense, and mission operations at the scale of a full network system. As discussed in Section 4, we leverage the re-usability of the hierarchical modeling to re-tool and re-use a network component model which has been used in two previous studies.[16,24]

Specifically, we utilize the network-scale *mission model* from these studies, which is based on a military-style Air Operations Center (AOC). The AOC mission is tasked with gathering requests for air operations and processing these into final flight plans. The mission model characterizes a network-supported, time-sensitive mission that allows us to examine a defensive mitigation's ability to protect the mission from attack. Any delay to the mission's completion is undesired. A mission team involves three mission users and three database servers existing on different network devices. We assume each mission device has at most one mission role. The abstracted AOC mission is shown in Figure 4, where the mission users pass a payload from Database 1 to Database 3. The network includes N_{m} mission user teams sharing three mission servers, meaning that there are $3N_{\text{m}} + 3$ total mission devices. Mission users require a fixed amount of uninterrupted time, t_{M}, to operate on the payload before passing it to the next step. Non-mission operations, such as benign communications can

Algorithm 2: Network attack/defense model

```
procedure EnclaveInit(encl)   ▷ Initialization of enclave encl
    $p_{dcomp} \leftarrow p_{dcomp}$ for encl from global params list   ▷ $p_{dcomp}$: probability of device compromise
    $t_{dcomp} \leftarrow t_{dcomp}$ for encl from global params list   ▷ $t_{dcomp}$: mean duration time of device compromise
    $\sigma_{tdcomp} \leftarrow \sigma_{tdcomp}$ for encl from global params list   ▷ $\sigma_{tdcomp}$: standard deviation of device compromise duration times
    devices ← [all devices in encl]
    for all device ∈ devices do
        $r \leftarrow \mathbb{N}[0,1]$   ▷ r is assigned a random value ∈ [0,1]
        if $r < p_{dcomp}$ then   ▷ device should be marked as compromised
            $\mu \leftarrow t_{dcomp}$
            $\sigma \leftarrow \sigma_{tdcomp}$
                comptime ← $\mathbb{N}(\mu,\sigma)$   ▷ Compute compromise duration time for device
                Mark device as compromised for time comptime
procedure Network   ▷ Run network-scale attack/defense
    $t \leftarrow 0$
    enclaves ← [all enclaves in network]
    for all encl ∈ enclaves do   ▷ Initialize devices in each network enclave
        EnclaveInit(encl)
    repeat
        $t \leftarrow t + 1$
        for all encl ∈ enclaves do
            devices ← [all devices in encl]
            for all device ∈ devices do
                if device compromise duration time is complete then
                    Mark device as uncompromised
                if all devices in encl are uncompromised then
                    EnclaveInit(encl)   ▷ Re-initialize devices in enclave encl
    until Total timesteps > Maximum timesteps
```

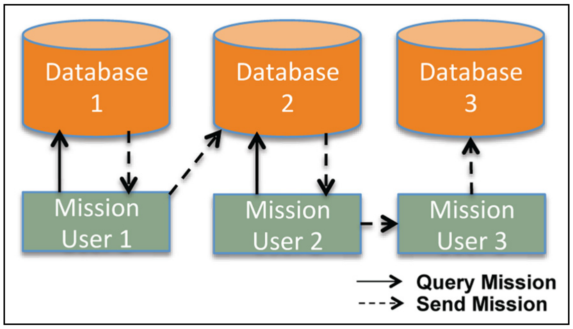


Figure 4. Abstracted AOC Mission Model: Three mission users utilize three network hosts to interact with three database servers to execute the mission.

occur during this time, but suffering a compromise to a mission device will delay the mission until that device is cleansed and restored.

The threat/defense model is an abstraction of the attack and defense dynamics captured in the enclave model but from a full network perspective where attack/defense outcomes vary depending on the micro-environment specified for individual enclaves in the full network. The model specifies a random variable to capture the probability that a device in a given network enclave is compromised. At simulation time $t = 0$, this variable is used to determine which devices in a given enclave are compromised and, for those that are compromised, a second random variable determines the duration of compromise. This initialization process is repeated separately for each enclave of the network. As simulation time progresses, compromised devices are cleansed and restored when their compromise durations have completed. After all compromised devices of an enclave have been restored, the initialization process is re-executed on the enclave, to set compromised devices and their corresponding compromise duration times. The full model of attack and defense at the network scale is given by Algorithm 2.

As discussed in the previous section, outputs from the enclave model are used to inform the network model. In Algorithm 2, the probability of device compromise, the mean compromise duration time for compromised devices, and the standard deviation of compromise duration times for a particular enclave are specified by the output of enclave model experiments for that enclave. Network model outputs measure the overall security and mission impact at the network scale. The following section provides details on the metrics used to measure these fundamental network concerns.

5 Measuring mitigation effectiveness

In this study we model attacks on device availability. A device becomes inaccessible after a successful attack and remains inaccessible until it is cleansed and restored. Our goal is to measure the effectiveness of a defensive mitigation with respect to system security and mission impact. For this purpose, we utilize the following two metrics as in Wagner et al.[24] and Priest et al.[16]

Definition 1. System Security Index, s_i *The expected ratio of device availability time (i.e., device uptime) to total time, normalized to* $[0, 1]$:

$$s_i = E\left[\frac{T_{\mathrm{up}}}{T}\right] = \frac{\sum_{i=1}^{3\times N_{\mathrm{m}}+3} \frac{T - t^i_{\mathrm{down}}}{T}}{3\times N_{\mathrm{m}}+3} \tag{2}$$

where $t^i_{\mathrm{down}} = t^i_{\mathrm{comp}} + t^i_{\mathrm{cleanse}}$ is the total downtime for device *i*, *T* is the total simulation time, and T_{up} is the total uptime for device *i* ($T_{\mathrm{up}} = T - T_{\mathrm{down}}$, $T_{\mathrm{down}} = \sum_{i=1}^{3\times N_{\mathrm{m}}+3} t^i_{\mathrm{down}}$). As discussed in Section 4, a device is inaccessible when it is compromised or while it is in the process of being cleansed. t^i_{comp} and t^i_{cleanse} represent the total compromise and cleansing times for device *i*, respectively.

Definition 2. Mission Delay, m_d *The expected total time of device compromise* $t_{(\mathrm{delay}|\mathrm{comp},\,d)}$ *and device cleansing* $t_{(\mathrm{delay}|\mathrm{cleanse},\,d)}$ *that impedes a mission.*

As detailed in Section 4.3, we model time-sensitive missions in which incurred delay is undesirable. When a mission-critical device *d* is compromised, the corresponding mission is delayed until the device has been cleansed. Mission delay is computed as:

$$m_{\mathrm{d}} = t_{(\mathrm{delay}|\mathrm{comp},\,d)} + t_{(\mathrm{delay}|\mathrm{cleanse},\,d)} \tag{3}$$

The expected total time of mission-impeding device compromise is computed as:

$$\begin{aligned} t_{(\mathrm{delay}|\mathrm{comp},\,d)} &= E\left[\hat{t}^i_{(\mathrm{delay}|\mathrm{comp},\,d)}\right] \\ &= \frac{\sum_{i=1}^{M} \hat{t}^i_{(\mathrm{delay}|\mathrm{comp},\,d)}}{M} \end{aligned} \tag{4}$$

where $\hat{t}^i_{(\mathrm{delay}|\mathrm{comp},\,d)}$ is the delay for mission *i* due to compromise and *M* is the number of executed missions.

The expected total time of mission-impeding device cleansing is computed as:

$$\begin{aligned} t_{(\mathrm{delay}|\mathrm{cleanse},\,d)} &= E\left[\hat{t}^i_{(\mathrm{delay}|\mathrm{cleanse},\,d)}\right] \\ &= \frac{\sum_{i=1}^{M} \hat{t}^i_{(\mathrm{delay}|\mathrm{cleanse},\,d)}}{M} \end{aligned} \tag{5}$$

where $\hat{t}^i_{(\mathrm{delay}|\mathrm{cleanse},\,d)}$ is the delay for mission *i* due to compromise.

5.1 A unified metric to evaluate mitigation effectiveness

The mitigations given in Section 3 exist to support an organizational mission and thus they should be evaluated in the context of the complete system where the goal is to maximize the system security index and to minimize the mission delay. These metrics evaluate different effects of a given mitigation. It is convenient to have a unified metric to evaluate the effectiveness of a mitigation and to compare the effectiveness of multiple mitigations.

Definition 3. Unified Metric, m_g. *It is a measure to characterize the security and mission delay (* s_i *and* m_{d} *from Equations (2) and (3), respectively) inherent to a simulated network environment captured via Monte Carlo experiments. The metric incorporates effects of mean, median, and variance of results from Monte Carlo simulation runs, normalized to [0,1].*

To generate this metric, we first unify the security index (Equation (2)) for Monte Carlo experiments of simulation scenarios with and without a given mitigation as shown below:

$$\left.\begin{aligned} s_{\mathrm{M}} &= \int f_2(f_1(s_{i,\,\mathrm{M}}))\,\mathrm{d}s_{i,\,\mathrm{M}} \\ s_{\mathrm{noM}} &= \int f_2(f_1(s_{i,\,\mathrm{noM}}))\,\mathrm{d}s_{i,\,\mathrm{noM}} \end{aligned}\right\} \Rightarrow s_g = \frac{s_{\mathrm{M}} - s_{\mathrm{noM}}}{\max(s_{\mathrm{M}}, s_{\mathrm{noM}})} \tag{6}$$

where $s_{i,\,\mathrm{M}}$ and $s_{i,\,\mathrm{noM}}$ represent the computed security index values for scenario experiments with and without the given mitigation, respectively. f_1 is a function $f_1 : X \rightarrow f_1{*}$ that takes an arbitrary input *X*, which might be $s_{i,\,\mathrm{M}}$ or $s_{i,\,\mathrm{noM}}$, and then outputs an approximation function $f_1{*}$. f_2 is a function $f_2 : f_1{*} \rightarrow f_2{*}$ that takes an arbitrary function $f_1{*}$, and then maps into an approximation function $f_2{*}$. s_{M} and s_{noM} are the approximated security index values with and without mitigation, respectively. s_g represents the unified security index, normalized to $[-1, +1]$. A computed value for $s_g \in [-1, 0)$ signifies that the proposed mitigation *decreases* overall security, while a computed value $\in (0, +1]$ signifies the proposed mitigation *increases* overall security.

Secondly, the mission delay (Equation (3)) for Monte Carlo experiments of scenarios with and without the given mitigation (m_{M} and m_{noM}, respectively) is unified by using the following equation:

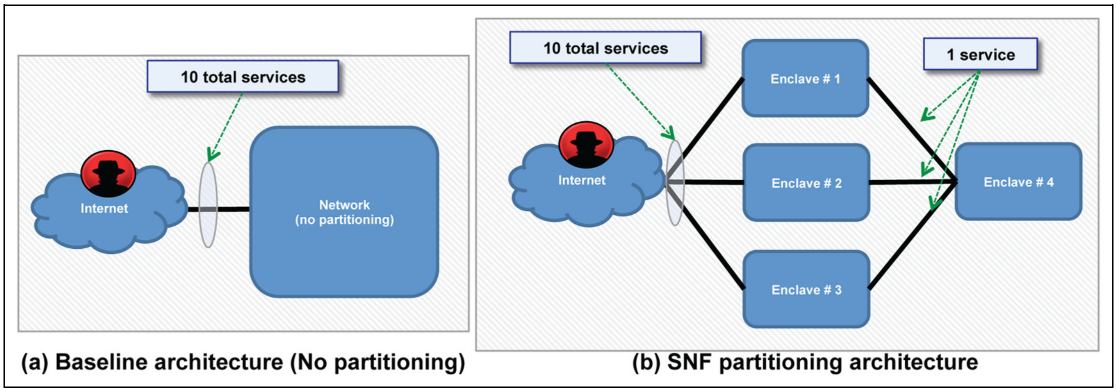


Figure 5. Baseline architecture and SNF's network partitioning architecture.

$$\left.\begin{aligned} m_{\mathrm{M}} &= \int f_4(f_3(md_{i,\,\mathrm{M}}))\,\mathrm{d}t \\ m_{\mathrm{noM}} &= \int f_4(f_3(md_{i,\,\mathrm{noM}}))\,\mathrm{d}t \end{aligned}\right. \Rightarrow md_g \quad = \frac{m_{\mathrm{noM}} - m_{\mathrm{M}}}{\max\left(m_{\mathrm{noM}}, m_{\mathrm{M}}\right)} \tag{7}$$

where $md_{i,\,\mathrm{M}}$ and $md_{i,\,\mathrm{noM}}$ represent the computed mission delay values for scenario experiments with and without the given mitigation, respectively. f_3 is a function $f_3 : X \to f_3{}^*$ that takes an arbitrary input X, which might be $md_{i,\,\mathrm{M}}$ or $md_{i,\,\mathrm{noM}}$, and then outputs an approximation function $f_3{}^*$. f_4 is a function $f_4 : f_3{}^* \to f_4{}^*$ that takes an arbitrary function $f_3{}^*$, and then maps into an approximation function $f_4{}^*$. m_{M} and m_{noM} are the approximated mission delays with and without mitigation, respectively. md_g represents the unified mission delay, normalized to $[-1, +1]$. A computed value for $m_g \in [-1, 0)$ signifies that the proposed mitigation *increases* mission delay, while a computed value for $m_g \in (0, +1]$ signifies the proposed mitigation *decreases* mission delay.

Finally, s_g and md_g are combined as:

$$m_g = f_5({}_w 1, s_g, w_2, md_g) \tag{8}$$

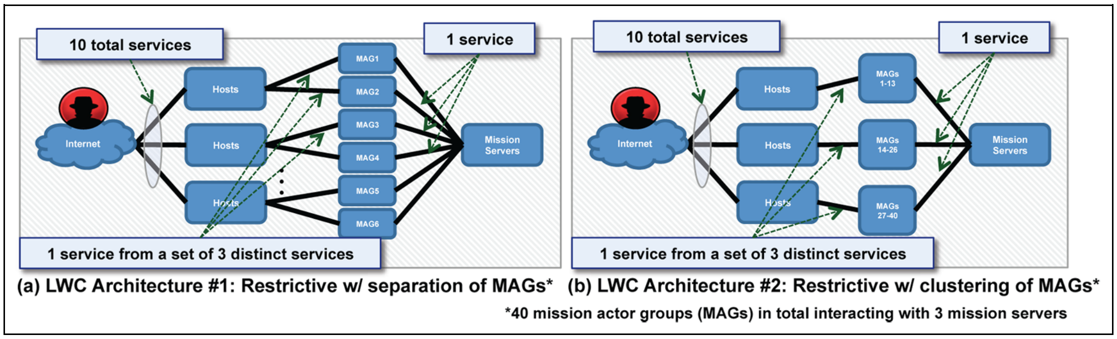
where f_5 is a function $f_5 : \{w_1, s_g, w_2, md_g\} \to m_g$ that inputs user-defined weighting factors, w_1 and w_2, that represent the relative importance of security and mission impact, respectively, to the user where $w_1 + w_2 = 1$, and the computed values of s_g and md_g and outputs the unified performance measure m_g.

The proposed effectiveness measure given in Equation (8) combines s_i and md_g to provide a unified metric for effectiveness. This metric can be used to measure the effectiveness of a single mitigation and/or compare the effectiveness of multiple mitigations. To the best of our knowledge, the measure given in Equation (8) represents the first attempt to unify the fundamental network concerns of system security and mission performance into a single metric that is easily accessible to security practitioners. The unified metric is also used in another study conducted by the authors [24] that was submitted at the same time as this study.

6 Experiments

The simulation framework utilized in this study is designed to model the environment, entities, and actors of cyber systems at relevant scales in order to gain a useful understanding of complex system dynamics. The goal is to understand sub-system dynamics, how these dynamics affect and are affected by the system. Our model utilizes this framework to capture a full network environment including users, attackers, defenders, and mission operations. The simulation framework is implemented using NetLogo.[25] Matlab release[26] 2014b and Python 2.7 are used for data aggregation across simulation runs and the calculation of statistical measures.

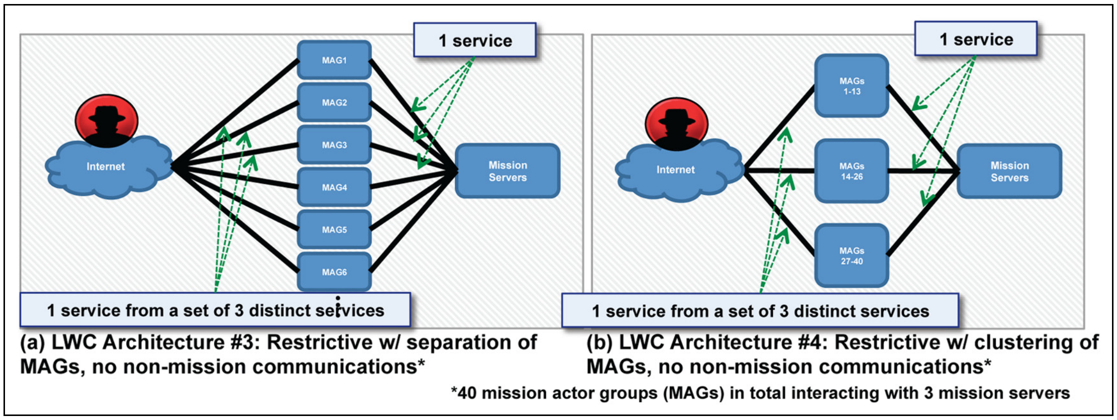
In this section, we analyze and quantify the effectiveness of two defensive mitigations: (i) SNF (Section 3.1); and (ii) LWC (Section 3.2). The goal of SNF and LWC is to partition a network into enclaves to restrict an attacker's ability to move in a network. As discussed in Section 4, we utilize a two-level hierarchical model that is informed by the outputs of testbed experiments. The testbed environment is used to test various network partitioning architectures as a function of communications between enclaves and between enclaves and the Internet. These communications are abstracted as information flows via software services (depicted in Figures 5–7 as black lines connecting enclaves/Internet; details provided in the following section). The first level of the modeling hierarchy, the *enclave model*, is meant to characterize device-to-device infection spreading within a single enclave. The second level is the *network model*, which is used to capture system security and mission impact. To capture these two fundamental



(a) LWC Architecture #1: Restrictive w/ separation of MAGs* (b) LWC Architecture #2: Restrictive w/ clustering of MAGs*

*40 mission actor groups (MAGs) in total interacting with 3 mission servers

Figure 6. LWC architectures that include both mission & non-mission communications.



(a) LWC Architecture #3: Restrictive w/ separation of MAGs, no non-mission communications*

(b) LWC Architecture #4: Restrictive w/ clustering of MAGs, no non-mission communications*

*40 mission actor groups (MAGs) in total interacting with 3 mission servers

Figure 7. LWC architectures that allow only mission communications.

concerns at the network scale, we model a representative network environment that supports the AOC mission described in Section 4.3.

6.1 Testbed experimental setup

As described in Section 4.1, our testbed is a proprietary environment that supports coarse-grained tests of a partitioned network. Although network partitioning *best practices* exist,[27] these provide only vague guidance and, thus, require significant interpretation by network administrators to implement. Generally, there exist many possible ways to implement network partitioning, and selection of an optimal partitioning architecture for a given network environment remains an open problem. For this study, we focus on examining representative partitioning architectures for a mid-sized organization. Here, we examine six partitioning scenarios: a baseline scenario in which no partitioning is used, an architecture representative of SNF, and four architectures representative of LWC. All of these scenarios include some form of Internet connectivity, which is modeled as one or more total services connecting the network to the Internet. Figures 5–7 depict the baseline, SNF, and the four LWC scenarios, respectively. The baseline scenario (Figure 5(a)) captures a network that is unpartitioned and the SNF scenario (Figure 5(b)) captures a coarsely-partitioned network with four enclaves that represent canonical organizational functions. Both of these architectures are connected to the Internet via 10 services, while the SNF scenario uses a single service to connect enclaves 1–3 to enclave 4. As detailed in Section 3.2, LWC provides a more finely-grained partitioning architecture than SNF and, thus allows more degrees of freedom with respect to the number of enclaves used and the restriction of communications between these enclaves. These extra degrees of freedom mean there are more possibilities to consider when choosing an architecture representative of the LWC mitigation. We, therefore, select four such architectures depicted in Figures 6(a) and (b) and 7(a) and (b). The first two, depicted in Figure 6, represent canonical separation of mission-centric and non-mission-centric communications while the last two, depicted in Figure 7, represent scenarios in which only mission-centric communications are present (see Section 6.3). Communications between enclaves and between enclaves and the Internet for these LWC scenarios are specified by services as shown in the figures.

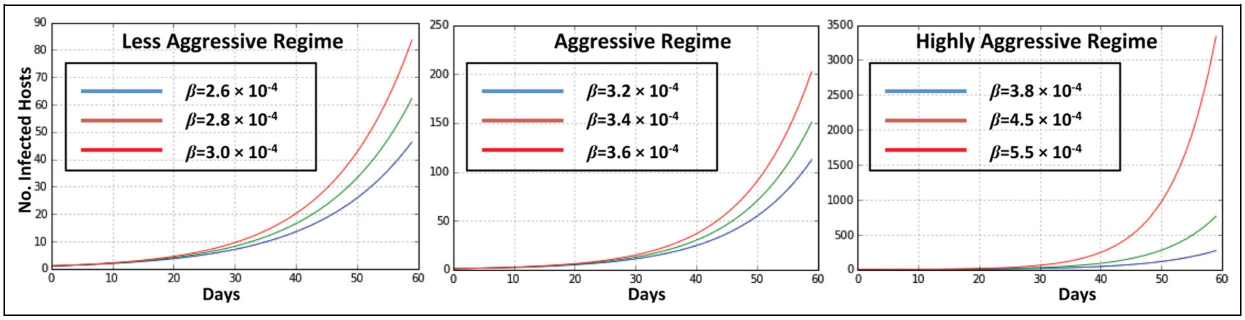


Figure 8. The impact of infection rate per unit time, β, on the spreading progress for a vulnerable network with $N = 250$ and $I(0) = 1$.

The output from testbed experiments provides the probability of enclave vulnerability for each enclave of a captured scenario.

6.2 Enclave model experimental setup

This model captures device-to-device infection spreading within an enclave. We assume that an attacker penetrates an enclave by compromising a single device and then attempts to spread to other devices in the enclave. As presented in Section 4.2, we use the model given in Yu et al.[18] to mimic infection spread in an enclave. The model inputs the probability of enclave vulnerability, p_{vuln}, for each enclave of a captured scenario from testbed experiments (see previous section). We model a class-C-sized network with 250 total hosts/devices.

Equation (1) is used to compute the number of compromised nodes at a given time where $N = 250$, $I(t = 0) = 1$ and three threat regimes with respect to infection spreading are examined. These regimes represent a differing severity of threat: (i) less aggressive spreading (low threat); (ii) aggressive spreading (medium threat); and (iii) highly aggressive spreading (high threat). As shown in Figure 8, the infection spread rate, β, is varied to model these different severities. To account for the uncertainty inherent to the spreading dynamic, we sweep a range of β values within each regime. Less aggressive spreading is given by $\beta = \{2.6 \times 10^{-4}, 2.8 \times 10^{-4}, 3.0 \times 10^{-4}\}$ which captures an attacker that can infect less than 40% of a class C network in 60 days. Aggressive spreading is given by $\beta = \{3.2 \times 10^{-4}, 3.4 \times 10^{-4}, 3.6 \times 10^{-4}\}$ and captures an attacker that can infect up to 80% of the network in 60 days. Finally, highly aggressive spreading is given by $\beta = \{3.8 \times 10^{-4}, 4.5 \times 10^{-4}, 5.5 \times 10^{-4}\}$ and models an attacker who can infect the entire network in less than 30 days.

6.3 Network model experimental setup

This model captures network-scale attack/defense dynamics and mission operations. As discussed above, we consider a representative class-C-sized network with 250 hosts. We model the AOC mission (described in Section 4.3) where the full mission takes three days to complete if uninterrupted and each of three mission users requires one day to complete his/her mission task. Simulation time is specified such that 1,000 time units = 1 simulated day. We collect results from 1,500 Monte Carlo simulation runs in which runs are terminated upon completion of all missions or when simulation time reaches a maximum of 30,000 time units (30 simulated days).

As described in Section 6.1, we examine six partitioning architectures: a baseline architecture (network with no partitioning, Figure 5(a)), a representative SNF architecture (partitioned with respect to canonical organizational functions, Figure 5(b)), and four representative LWC architectures (Figures 6 and 7). The LWC architectures represent two general scenarios, one in which a mix of mission and non-mission communications are allowed, and one in which only mission communications are allowed.

Figure 6 depicts two representative architectures allowing both mission and non-mission communications but separate mission-critical devices from non-mission-critical devices. In both of these architectures, enclaves labeled *Hosts* contain non-mission devices and enclaves labeled MAG1, MAG2, etc. contain mission-critical devices used by mission actor teams or groups (abbreviated as MAGs in the figure—see Section 4.3 for discussion) that communicate with mission servers in the enclave labeled *Mission Servers*. The difference between these two architectures is in how MAGs are separated: in LWC Architecture #1 (Figure 6(a)) MAGs are completely separated, where each MAG is contained in its own enclave, while in LWC Architecture #2 (Figure 6(b)) MAGs are clustered into three MAG-only enclaves. Figure 7 depicts two representative architectures that allow only mission-critical devices. In both of the architectures, the layer of host enclaves is removed so that only MAG and the mission server enclaves remain. LWC Architecture #3 (Figure 7(a)) mirrors Architecture #1 with the Host enclaves removed,

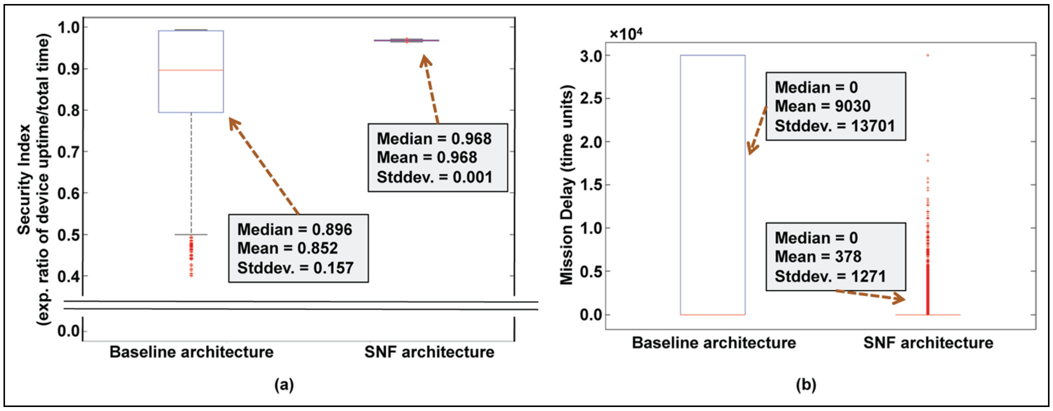


Figure 9. SNF simulation results: (a) security index, s_i ; and (b) mission delay, m_d.

while LWC Architecture #4 (Figure 7(b)) mirrors Architecture #2 also with the Host enclaves removed.

6.4 Simulation results

The following sections present simulation results for the partitioning scenarios described above. First, SNF scenario simulation results are compared to baseline scenario results and then LWC scenario results are compared to baseline scenario results. The purpose here is to examine the relative benefit offered by each mitigation with respect to the baseline (i.e., no mitigation).

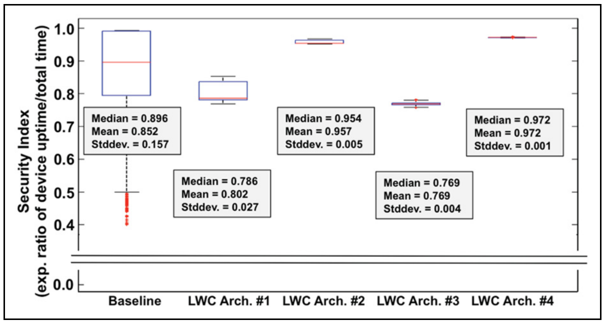


Figure 10. LWC simulation results for security index, s_i.

6.4.1 SNF results. We compare two scenarios to analyze the effectiveness of the SNF mitigation. A baseline scenario in which no partitioning is used, as shown in Figure 5(a) and a representative SNF architecture in which the entire network is subdivided into four enclaves as shown in Figure 5(b).

Simulations are executed for both scenarios and output metrics are computed. Figure 9 shows the computed metrics s_i and m_d of Equations (2) and (3), respectively, visualized as statistical box plots depicting the median metric level (red line in the figure) and the variance of computed values over the 1,500 Monte Carlo experiments. As seen in Figure 9(a), the SNF partitioning architecture has a higher s_i on average, with a mean expected ratio of device uptime of 0.968 as opposed to 0.852 for the baseline case. Furthermore, results also indicate that there is significantly less variance in security performance for the SNF architecture, with a standard deviation of 0.001 as opposed to 0.157 for the baseline architecture. This result is compelling as it is a reduction in variance of two orders of magnitude. Figure 9(b) shows significant improvements to mission impact. The SNF architecture gives both lower average mission delay and lower variance in mission performance. The average mission delay for the SNF architecture is 378 time units as opposed to 9,030 time units for the baseline architecture (an order-of-magnitude difference) while the standard deviation is 1, 271 time units for SNF as opposed to 13,701 time units for the baseline (also an order-of-magnitude difference).

6.4.2 LWC results. To analyze the effectiveness of the LWC mitigation, we compare five scenarios: the baseline architecture (Figure 5(a)) and four representative LWC architectures (Figures 6 and 7). Simulations are executed for all five scenarios and output metrics are computed. Results are given in Figures 10 and 11.

As seen in Figure 10, results for the security index, s_i, show that LWC yields marked improvements to security. All of the LWC architectures give significantly less variance in security performance relative to the baseline scenario. Two of the four of the LWC architectures give higher average security performance relative to the baseline architecture, while for the other two architectures, the average security is lower but comparable to that of the baseline. The architecture with the best result, LWC

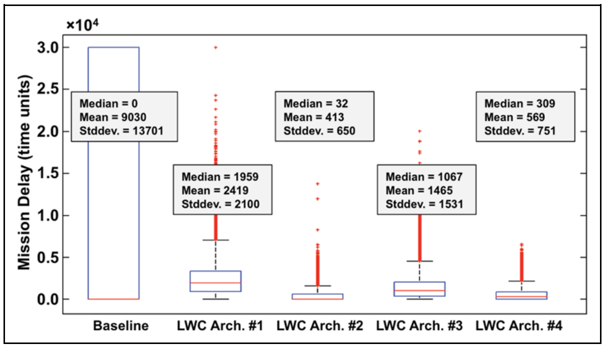


Figure 11. LWC simulation results for mission delay, m_d.

Architecture #4, yields a mean expected ratio of device uptime of 0.972 as opposed to 0.852 for the baseline architecture. The four LWC architectures yield a variance in security performance ranging from 0.027 to 0.001 compared to 0.157 for the baseline architecture. This is a compelling result as it indicates a reduction in variance of one to two orders of magnitude relative to the baseline.

As given in Figure 11, simulation results show that LWC yields noticeable improvements in mission impact. All LWC architectures have lower average mission delay and lower variance in mission delay relative to the baseline architecture. Mean mission delay ranges from 2, 419 to 413 time units for the LWC architectures, while for the baseline it is 9,030 time units. Variance in mission delay ranges from 2,100 to 650 time units for the LWC architectures compared to the baseline, which is 13, 701 time units. These results are also quite compelling as they indicate an improvement of approximately one order of magnitude for mean mission delay and one to two orders of magnitude in mission delay variance.

It is important to note that reported results for the baseline are affected by the maximum simulation running time (30,000 time units). For many simulation runs of the baseline architecture, missions did not complete before this time limit and mission delay was therefore computed as the maximum value of 30,000. Thus, the difference in mission performance between the examined mitigation scenarios (SNF and LWC) and the baseline scenario may be even more striking than reported here.

6.5 Unified metric computation

In Section 5.1, we introduced a unified measure for comparing the effectiveness of various mitigations. The goal of m_g is to provide a single measure for effectiveness as explained in Definition 3. The proposed approximation functions f_1, f_2, f_3 and f_4 given in Equations (6) and (7) are general functions mapping simulation experiment results into functions that are integrable. f_5 shown in Equation (8) takes the approximated security index and mission delay values and combines them with the importance factors to generate single evaluation value. To incorporate both mean and variance into m_g, we use the following functions:

$$
\begin{aligned}
f_1 &= \bigoplus_{i=1}^{\infty} N_i \times (s_{i,\mathrm{M}} \vee s_{i,\mathrm{noM}}) \rightarrow \bigoplus_{k=1}^{n} s_k = S_k \\
f_2 &= \frac{1}{n} \times \sum_{i=1}^{n} (S_{k,i}) \\
&\quad \times \mathcal{N}\left\{\frac{1}{n} \times \sum_{i=1}^{n} S_{k,i}, \mathbb{E}\left(\sum_{i=1}^{n} (S_{k,i}) - \frac{1}{n} \times \sum_{i=1}^{n} (S_{k,i})\right)^2\right\} \\
f_3 &= \bigoplus_{i=1}^{\infty} N_i \times (md_{i,\mathrm{M}} \vee md_{i,\mathrm{noM}}) \rightarrow \bigoplus_{k=1}^{n} md_k = Md_k \\
f_4 &= \tfrac{1}{n} \times \sum_{i=1}^{n} (Md_{k,i}) \\
&\quad \times \mathcal{N}\left\{\tfrac{1}{n} \times \sum_{i=1}^{n} Md_{k,i},\ \mathbb{E}\left(\sum_{i=1}^{n} (Md_{k,i}) - \tfrac{1}{n} \times \sum_{i=1}^{n} (Md_{k,i})\right)^2\right\} \\
f_5 &= m_g = 0.5 \times s_g + 0.5 \times md_g
\end{aligned}
\tag{9}
$$

where $\oplus_{i=1}^{\infty} N_i$ represents all possible histograms and n is the total number of measurements. f_1 and f_3 take the security index and the mission delay simulation results for the no mitigation, SNF, and LWC scenarios and map them into histograms. f_2 and f_4 fit the Gaussian distribution to each newly created histogram and then multiply by the mean of each measurement. f_5 is a linear function averaging both the enhanced security index and mission delay.

Figure 12 presents S_k results (shown in black) with the Normal distribution approximation function (shown in red) for all scenarios. The histogram approximations are obtained by applying f_1 (see Equation (9)) on the simulation outputs. f_2 presented in Equation (9) is another approximation function that takes histogram approximations and fits these to the Normal distribution.

Note that, as shown in Figure 12, simulation runs for LWC Architectures #1 and #2 exhibit bimodal distributions for S_k. This is due to the extra layer of Host enclaves specified by the architectures in Figure 6: attacks that successfully penetrate the network and make it past the layer of Host enclaves cause a noticeable drop in overall security performance, while attacks that do not make it past this first layer of partitioning result in noticeably better overall security. These outcomes specify the two modes of the distribution.

Now, we calculate the area under each red curve to compute the approximated security index for each architecture shown in Figures 5–7.

Note that the Normal distribution is a symmetric curve. To reward the simulation results at the right side of the curve (close to 1 for s_i), the enhanced security index in Equation (6) can be adjusted with the mean of each experiment and calculated as:

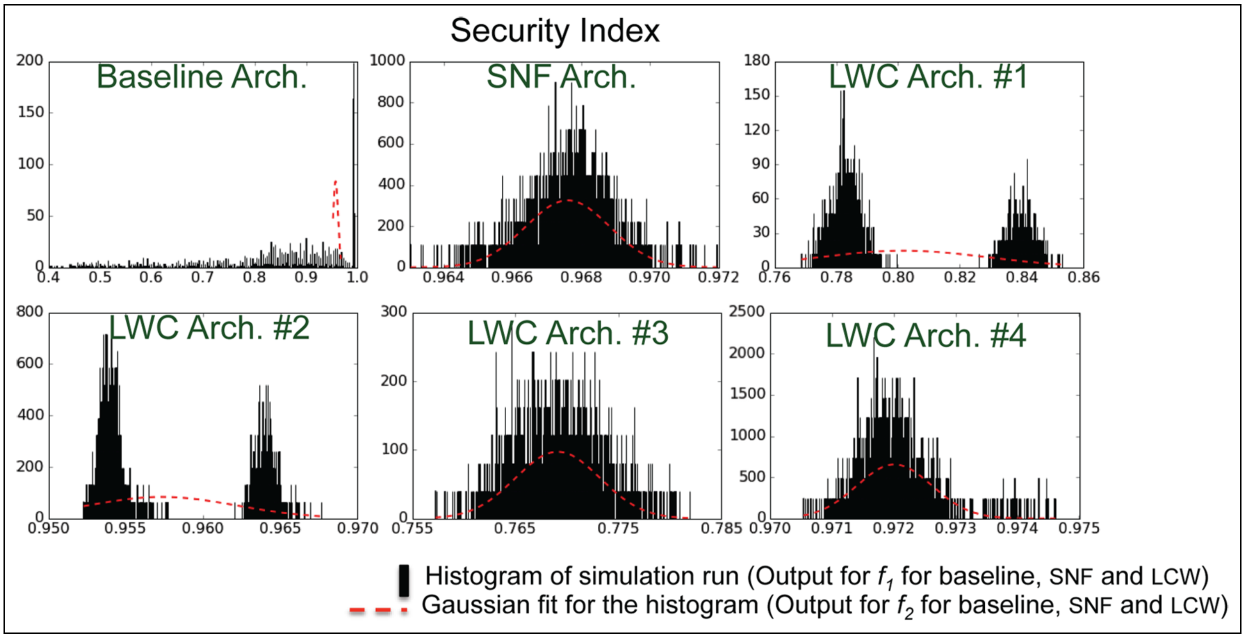


Figure 12. S_k (Equation (9)) results for all scenarios (axes are displayed with differing scales to improve readability).

Table 1. m_g for all scenarios.

Scenario	s_g	md_g	m_g
SNF	0.322	0.944	0.63
LWC 1	0.037	0.657	0.34
LWC 2	0.185	0.951	0.56
LWC 3	0.13	0.804	0.46
LWC 4	0.318	0.928	0.62

$$s_g = \frac{(s_{\mathrm{M}} \times \mathbb{E}(s_{\mathrm{M}}) - s_{\mathrm{noM}} \times \mathbb{E}(s_{\mathrm{noM}}))}{\max\,(s_{\mathrm{M}} \times \mathbb{E}(s_{\mathrm{M}}), s_{\mathrm{noM}} \times \mathbb{E}(s_{\mathrm{noM}}))} \qquad (10)$$

Based on Equation (10), the enhanced security indexes for all scenarios are shown in Table 1.

Figure 13 presents Md_k results (shown in blue) with the Normal distribution approximation function (shown in red) for all scenarios. The histogram approximations of each mitigation's mission delay results are obtained by applying f_3 (see Equation (9)) on the results. f_4 presented in Equation (9) is another approximation function taking histogram approximations and fitting these to the Normal distribution. As can be seen in the figure, many runs of the baseline architecture have a mission delay of 30,000 time units. This means these missions were not completed before the maximum simulation run time. When we run our experiments with a larger maximum run time, the number of uncompleted missions decreases; however, this does not result in significant changes to the computed value of f_4* (detailed in Section 5.1).

It is also interesting to note that simulation runs for LWC Architectures #1 and #2, in Figure 13, do not exhibit bimodal distributions for Md_k, as is seen for the S_k results (Figure 12) for these same architectures. This is due to the uncertainty inherent to mission operations and attacks on the mission. When an attack manages to penetrate the network and get past the first layer of Host enclaves for these architectures, it is still not certain it will be able to negatively impact the mission. Due to chance, the attack may compromise mission-critical devices that have already completed their mission operations and, thus, no mission delay will result. This dynamic prevents the distribution from being bimodal.

Due to the similar symmetry of the Normal distribution, we also add a reward factor into Equation (7) and the enhanced mission delay is calculated as:

$$md_g = \frac{(m_{\mathrm{noM}} \times \mathbb{E}(m_{\mathrm{noM}}) - m_{\mathrm{M}} \times \mathbb{E}(m_{\mathrm{M}}))}{\max\,((m_{\mathrm{noM}} \times \mathbb{E}(m_{\mathrm{noM}}), m_{\mathrm{M}} \times \mathbb{E}(m_{\mathrm{M}})))} \qquad (11)$$

Based on Equation (11), the enhanced mission delays for all scenarios are given in Table 1.

Assume that s_g and md_g are equally important concerns with respect to mitigation effectiveness and, thus, w_1 and w_2 of Equation (8) are both specified as 0.5. The unified performance metric m_g for SNF and LWC mitigations are shown in Table 1. From a practical standpoint, network administrators can view these results as a measurement of the gain in effectiveness at the network scale due to the mitigation. From the table, SNF yields a 63% gain in

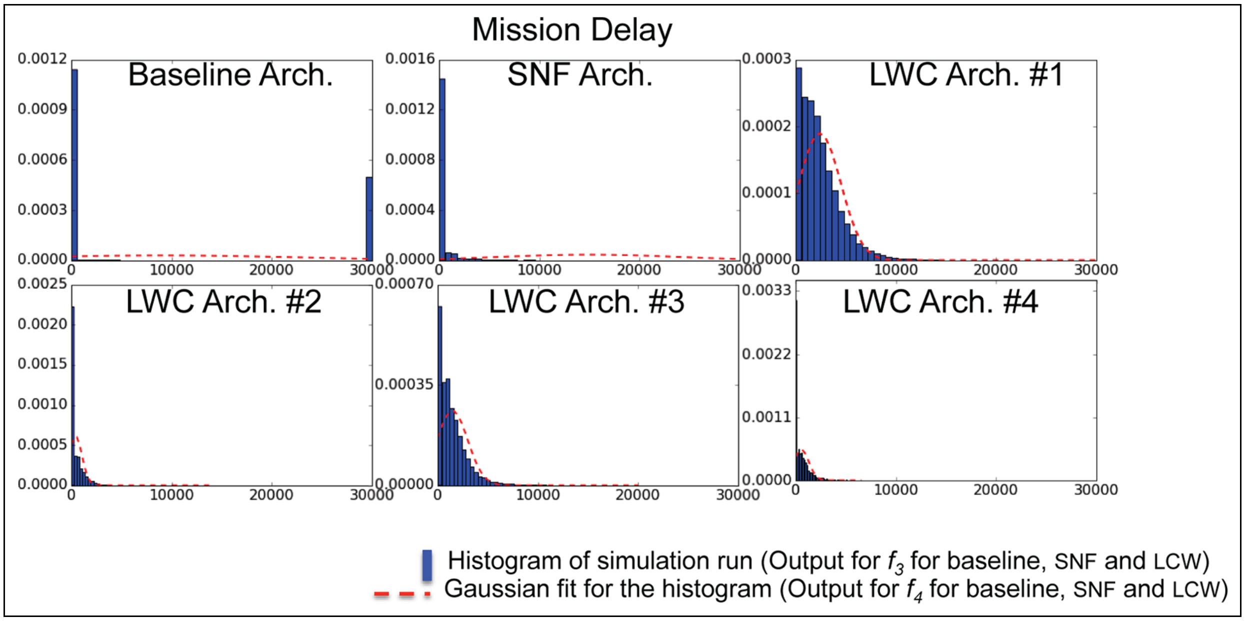


Figure 13. Md_k (Equation (9)) results for all scenarios (axes are displayed with differing scales for readability).

effectiveness while LWC yields gains ranging from 34% to 62% depending on the architecture implemented. These results indicate that both the SNF and LWC mitigations offer significant benefits to the security posture of a network. Also, the range of results seen for the four LWC architectures examined, show that although LWC has the potential to be a highly effective defensive mitigation, it can be quite difficult to select an appropriate partitioning architecture to implement this mitigation. Thus, network administrators should exercise caution when deploying LWC, as sub-optimal selection of partitioning architecture can have deleterious results on network security. It is also important to note that defensive mitigations are not meant to be used in isolation, but rather in combination as part of a layered defense. Thus when utilized as part of a greater defensive policy, SNF and LWC can provide great contributions to the security posture of a network against attack.

7 Conclusion

This paper presents a multi-scale hierarchical simulation model designed to evaluate two well-known network-level cyber defense mitigations, SNF and LWC. We quantify the network-scale effects of these mitigations from the perspectives of security and mission impact. Experimental results indicate that both mitigations provide significant benefits to the security posture of a network, with the caveat that LWC results can vary widely due to the extra degrees of freedom involved in selecting an appropriate architecture to implement it. We also introduce a novel metric that combines results for security and mission performance into a single unified measure of mitigation effectiveness that is convenient and easily accessible to security practitioners. This measure can be viewed by practitioners as a quantification of the gain in effectiveness at the network scale due to a defensive mitigation.

Future work is focused on developing a simulation model to replace the testbed, so that more partitioning scenarios can be easily examined, as the resource cost of executing scenarios on the testbed is relatively high. We also plan to test the inclusion of new functions to improve our unified measure.

Funding

This material is based upon work supported under US Air Force Contract No. FA8721-05-C-0002 and/or FA8702-15-D-0001. Any opinions, findings, conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the US Air Force.

References

1. Bennett C. Study: Cyberattacks up 48 percent in 2014. http://thehill.com/policy/cybersecurity/221936-study-cyber-attacks-up-48-percent-in-2014 (2014, accessed 1 August 2016).
2. CYREN cyber threats yearbook. Technical report, CYREN Corporation, USA, December 2015.
3. Barret D. US suspects hackers in China breached about four (4) million people's records, officials say. *Wall Street Journal*, 5 June 2015.

4. Zetter K. Sony got hacked hard: What we know and don't know so far. https://www.wired.com/2014/12/sony-hack-what-we-know/ (2014, accessed 1 August 2016).
5. Center for Internet Security. Critical security controls for effective cyber defense version 6.0. Technical report, SANS Institute, USA, October 2015.
6. Google's approach to IT security. Technical report, Google, USA, January 2012.
7. Enterprise security best practices. Technical report, Microsoft, USA, December 2015.
8. Top 10 information assurance mitigation strategies. Technical report, Information Assurance Directorate, NSA, USA, July 2013.
9. Shakshuki EM, Kang N and Sheltami TR. EAACK – a secure intrusion-detection system for MANETs. *IEEE Transactions on Industrial Electronics* 2013; 60: 1089–1098.
10. Nadeem A and Howarth M. Protection of MANETs from a range of attacks using an intrusion detection and prevention system. *Telecommun Syst* 2013; 52: 2047–2058.
11. Cheng C, Tay WP and Huang GB. Extreme learning machines for intrusion detection. In: *The 2012 International Joint Conference on Neural Networks (IJCNN)*, Brisbane, Australia, 10–15 June 2012, pp.1–8. Piscataway, NJ: IEEE.
12. Kiesling E, Strauss C, Ekelhart A, Grill B and Stummer C. Simulation-based optimization of information security controls: an adversary-centric approach. In: *2013 Winter Simulations Conference (WSC)* Washington DC, USA, 8–11 December 2013, p.2054–2065. IEEE.
13. Kotenko I, Konovalov A and Shorov A. Agent-based simulation of cooperative defence against botnets. *Concurr Comput* 2012; 24: 573–588.
14. Masi D, Fischer MJ, Shortle JF and Chen CH. Simulating network cyber attacks using splitting techniques. In: *Proceedings of the Winter Simulation Conference*, Pheonix, USA, 11–14 December 2011, pp.3212–3223. IEEE.
15. Malekzadeh M, Ghani AAA, Subramaniam S and Desa J. Validating Reliability of OMNeT in Wireless Networks DoS Attacks: Simulation vs. Testbed. *Int J Network Security* 2011, 3: 13–21.
16. Priest BW, Vuksani E, Wagner N, Tello B, Carter KM and Streilein WW. Agent-based simulation in support of moving target cyber defense technology development and evaluation. In: *Proceedings of the 18th Symposium on Communications & Networking*, San Diego, USA, April 2015, pp.16–23. Society for Computer Simulation International.
17. Toutonji OA, Yoo SM and Park M. Stability analysis of VEISV propagation modeling for network worm attack. *Appl Math Model* 2012; 36: 2751–2761.
18. Yu S, Gu G, Barnawi A, Guo S and Stojmenovic I. Malware propagation in large-scale networks. *IEEE Trans Knowledge Data Eng* 2015; 27: 170–179.
19. Wagner N, Lippmann R, Winterrose M, Riordan J, Yu T and Streilein WW. Agent-based simulation for assessing network security risk due to unauthorized hardware. In *Proceedings of the Symposium on Agent-Directed Simulation* April 2015, pp. 18–26. Society for Computer Simulation International.
20. Clark A, Sun K, Bushnell L and Poovendran R. A Game-Theoretic Approach to IP Address Randomization in Decoy-Based Cyber Defense. In: *International Conference on Decision and Game Theory for Security*, November 2015, pp. 3–21. Springer International Publishing.
21. Pawlick J, Farhang S and Zhu Q. Flip the Cloud: Cyber-Physical Signaling Games in the Presence of Advanced Persistent Threats. In *International Conference on Decision and Game Theory for Security*, November 2015, pp. 289–308. Springer International Publishing.
22. Gezelter R. Security on the internet. In: Hutt AE, Bosworth S and Hoyt DB (eds) *Computer security handbook*. 3rd ed. Wiley, 1995, pp.23-2–23-5
23. Limiting workstation-to-workstation communication. Technical report, Information Assurance Directorate, NSA, USA, July 2013.
24. Wagner N, Şahin C, Hanson D, Pena J, Vuksani E and Tello B. Quantitative analysis of the mission impact for host-level cyber defensive mitigations. In: *Proceedings of the 49th Annual Simulation Symposium*, San Diego, USA, April 2016, p.2. Society for Computer Simulation International.
25. Netlogo (version 5.1.0), http://ccl.northwestern.edu/netlogo/ (2015, accessed 1 August 2016).
26. Matlab (version 2014), http://www.mathworks.com/products/matlab/(2014, accessed 1 August 2016).
27. Reichenberg N. Improving security via proper network segmentation, http://www.securityweek.com/improving-security-proper-network-segmentation (2014, accessed 1 August 2016).

Author biographies

Neal Wagner holds a PhD degree in information technology and an MS degree in computer science, both from the University of North Carolina at Charlotte, and a BA degree in mathematics from the University of North Carolina at Asheville.

Dr Neal Wagner is a technical staff member in the Cyber Analytics and Decision Systems Group at MIT Lincoln Laboratory. His research focus lies in developing and applying computational intelligence techniques for cyber applications of modeling and simulation, optimization, and prediction. He has both academic and industrial experience. Prior to joining the laboratory in 2013, he was at SolveIT Software, where he specialized in the commercialization of bio-inspired computing techniques for supply-chain optimization of large organizations. His academic experience includes stints as a faculty member of the Computer Science and Information Systems Departments at Fayetteville State University and Augusta University.

Cem Şafak Şahin received his BS degree from Gazi University, Turkey in 1996, MS degree from Middle East Technical University, Turkey in 2000, and MPhil and PhD degrees from the City University of New York in 2010, all in Electrical Engineering.

Currently, he is a technical staff member in the Cyber Analytics and Decision Systems Group at MIT Lincoln

Laboratory. Before joining the laboratory in 2014, he was Senior Research Engineer at BAE Systems–AIT (formerly Alphatech) in Burlington, Massachusetts. Prior to that, he was a Senior Software Engineer at Elanti Systems in New Jersey. He was also a Principal Engineer in systems design at MiKES Inc., a Turkish defense company specializing in electronic warfare systems; in this role, he worked as part of a multi-national defense project in the United States and as an Engineer at Roketsan Inc.

His interests include wireless ad hoc networks, evolutionary algorithms, communication theory, multi-sensor fusion, algorithm development, artificial intelligence, machine learning, cybersecurity and modeling, and electronic warfare systems.

Michael Winterrose received his BS degree, magna cum laude, in mathematical physics from Washington State University in 2004, and his MS and PhD degrees in computational materials science from the California Institute of Technology in 2007 and 2011, respectively.

Dr Winterrose is a technical staff member in the Cyber Analytics and Decision Systems Group at MIT Lincoln Laboratory. He joined the laboratory in 2010 and has been developing models of cyber operations and adversarial dynamics using techniques from game theory, agent-based modeling, and artificial intelligence. Prior to joining the Cyber Analytics and Decision Systems Group, Dr Winterrose worked for two years as a systems analyst in the Intelligence, Surveillance, and Reconnaissance Systems and Technology Division at Lincoln Laboratory. His research interests include advanced simulation techniques, statistical analysis, and complex systems modeling.

James Riordan received a PhD degree in mathematics from the University of Minnesota in 1997. During his studies, he was a member of the Architecture, Research, and Technology group of the Secure Computing Corporation and a consultant to Counterpane Internet Security.

Dr Riordan is a member of the technical staff in the Cyber Systems Assessment Group. He joined MIT Lincoln Laboratory in June 2009. His research interests include operational security, applied cryptography, risk assessment, resilient computing, and the semantic web. Prior to joining the Laboratory, he was a research staff member at the IBM Research Laboratory in Zurich, Switzerland, for 12 years; at this laboratory he led numerous security-related projects ranging from mobile computing to intrusion detection to web-centric trust enhancement, and was named an IBM Master Inventor. He served on the executive board of the resilient computing network of excellence of the European Union.

Diana Hanson holds an MPS degree in information science from Pennsylvania State University and a BS degree in physics and astronomy from the University of Maryland.

Currently, she is an assistant technical staff member at MIT Lincoln Laboratory. She transferred to the Cyber Analytics and Decision Systems Group in August 2015. Her current work is in cyber applications of modeling and simulation. Her current research interests include agent-based modeling, machine learning, and decision support.

Jaime Peña earned BS degrees in computer science (2014) and applied mathematics (2015) from the University of Texas at El Paso (UTEP). His undergraduate research focused on analyzing gene patterns derived from single-nucleotide polymorphism in RNA sequences. Prior to joining MIT Lincoln Laboratory, he worked developing web-based applications used at UTEP and the Johns Hopkins Applied Physics Laboratory.

Mr Peña is an assistant staff member in the Cyber Analytics and Decision Systems Group at MIT Lincoln Laboratory. Since he joined the laboratory in August 2015, he has worked on the analysis of cyber systems using simulations and models. His research interests are in data analysis, cybersecurity, machine learning, and bioinformatics.

William W Streilein holds a BA degree in mathematics from Austin College, an MM degree in electronic and computer music from the University of Miami, and a PhD degree in cognitive and neural systems from Boston University. He is a senior member of the IEEE. He has been at Lincoln Laboratory since 1998.

Dr Streilein is the Leader of the Cyber Analytics and Decision Systems Group at MIT Lincoln Laboratory, where he manages research and development programs in cyber security. His current research interests include the application of machine learning and modeling techniques to problems in cyber security. Other areas of interest include the investigation and development of quantitative metrics for cyber security, the exploration of moving target techniques to improve the resiliency of cyber and cyber-enabled systems, and automated techniques for discovering how government missions map to cyber infrastructure and for assessing risk to mission systems. Prior to joining the his current group, Dr Streilein was a technical staff member in the Sensor Exploitation Group, where his research focused on the exploitation of multi-sensor fused imagery in interactive automatic learning and recognition environments.